Run Video Chat within your Unity application 📱🎮 (iOS)

When playing games with your friends, the real fun comes when you can trash-talk and gloat on how good you are! Whether you are playing a first-person shooter or playing a game of poker, you want to be able to communicate with your team or see the competitors' reactions to see if they are bluffing. More mobile games are adding real-time communication within their app to increase user stickiness and user engagement.

The Agora Video SDK for Unity enables you to add in-game real-time communications to increase interactivity between players with minimum impact on the gameplay.

A sample app is included in the Unity Asset and demonstrates the basic Agora SDK features:

- Join a Channel
- Leave a Channel
- Video Chat with multiple participants

This blog post will enable you to run the sample app on an iOS device within a few minutes.

. . .

Prerequisites

- Agora.io Developer Account

- Unity 3D 5.5+

Get an Agora Developer Account & App ID

To build and run the sample application you must obtain an App ID:

1. Create a developer account at agora.io. Once you finish the signup process, you will be redirected to the Dashboard.

2. Navigate in the Dashboard tree on the left to **Projects > Project List**.

3. Copy the App ID that you obtained from the Dashboard into a text file. You will use this when you launch the app.

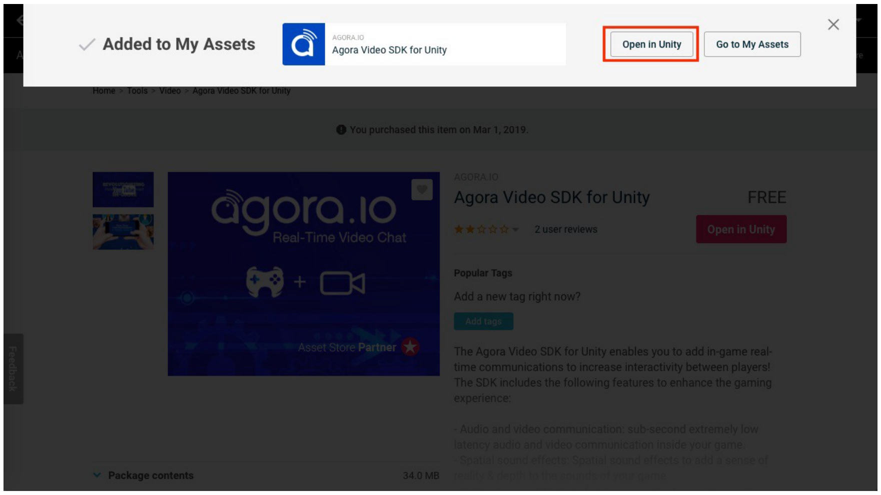
. . .

Set up the Demo Application

1. Add Agora Video SDK for Unity to your assets from the Unity Asset Store.

Note: If you see a **Terms of Service** page, click **Accept** to add the project.
Once the project has been added, a confirmation message will appear at the top of the screen.

2. Click **Open in Unity** to launch Unity.

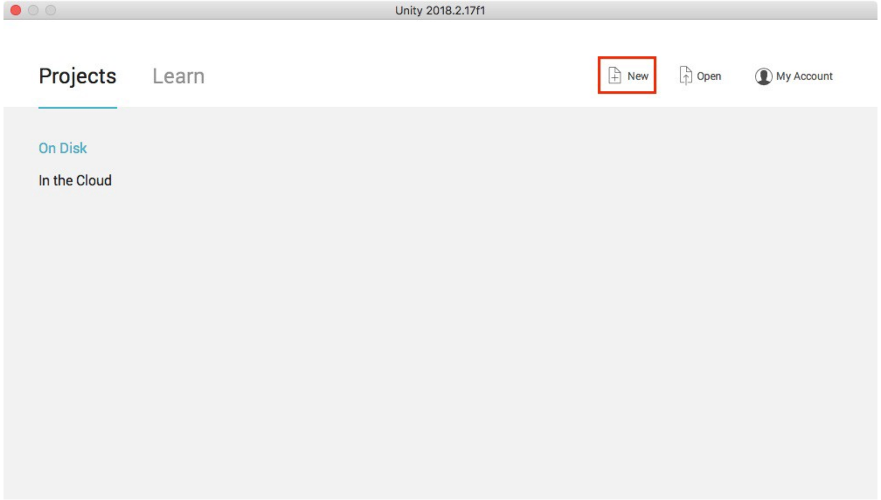


You purchased this item on Mar 1, 2019.

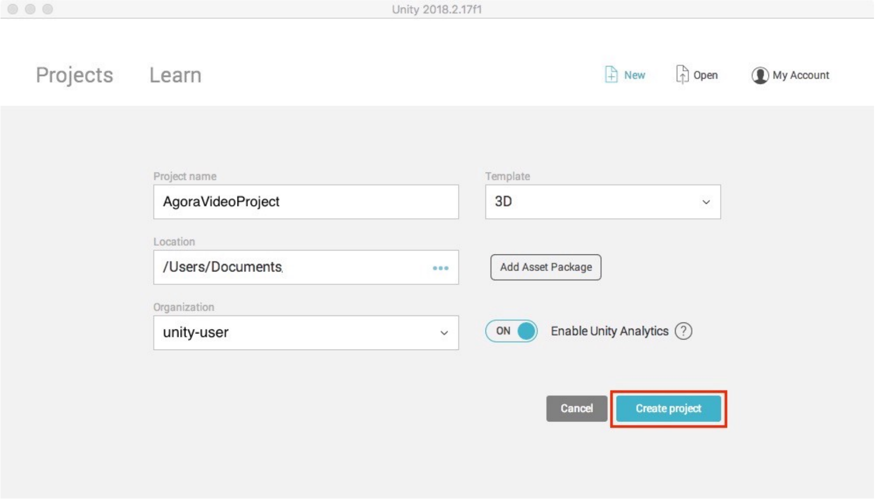
The Agora Video SDK for Unity enables you to add in-game real-time communications to increase interactivity between players! The SDK includes the following features to enhance the gaming experience:

- Audio and video communication: sub-second extremely low latency audio and video communication inside your game.
- Spatial sound effects: Spatial sound effects to add a sense of reality & depth to the sounds of your game.

3. When Unity opens, click **New** to start a new project.



4. Click **Create project** to create a new project.



Project name: AgoraVideoProject

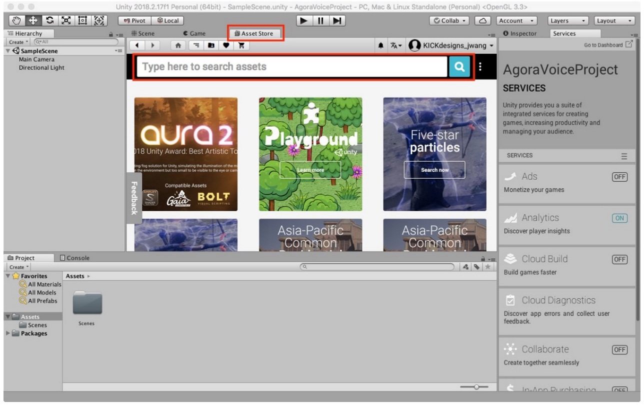
Template: 3D

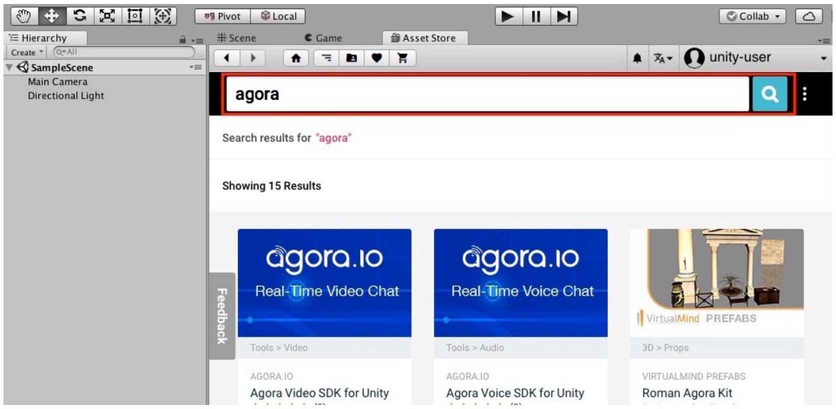
Location: /Users/Documents

Organization: unity-user

ON Enable Unity Analytics

5. Once your project is created, ensure the **Asset Store** tab is selected. If the Agora Video project is not already loaded, search for **Agora** in the search box.





Search results for "agora"

Showing 15 Results

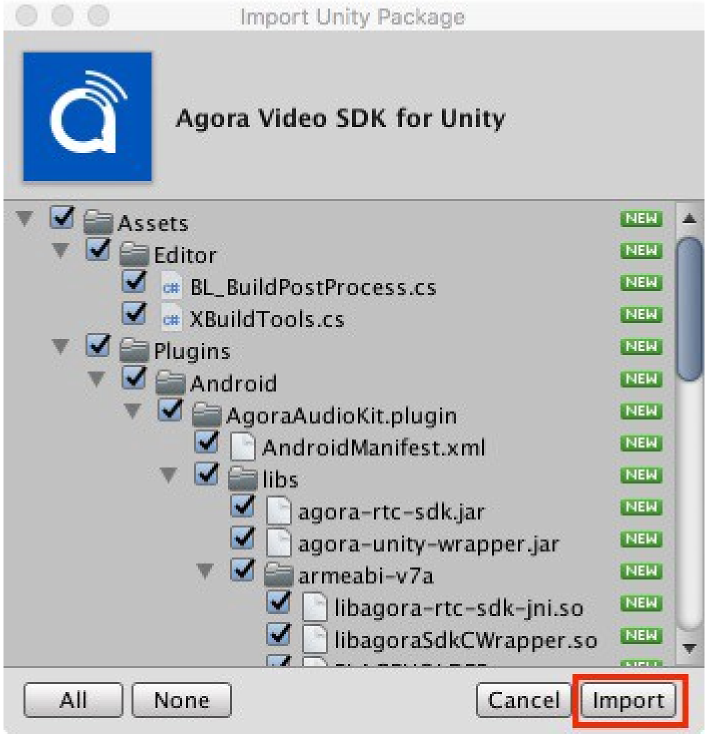
Find the **Agora Video SDK for Unity** project. Click the project box to open the project details.

6. Scroll down and click **Download**.

Note: The button changes to a loading bar to show download progress.

7. When the download is complete, the **Import** button will appear. Click **Import** to load Agora Video into your project.

8. Scroll through the asset list to familiarize yourself with the project structure and click **Import** to add the Agora.io assets into your current project.



Import Unity Package

Agora Video SDK for Unity

- ☑ Assets NEW
  - ☑ Editor NEW
    - ☑ BL_BuildPostProcess.cs NEW
    - ☑ XBuildTools.cs NEW
  - ☑ Plugins NEW
    - ☑ Android NEW
      - ☑ AgoraAudioKit.plugin NEW
        - ☑ AndroidManifest.xml NEW
        - ☑ libs NEW
          - ☑ agora-rtc-sdk.jar NEW
          - ☑ agora-unity-wrapper.jar NEW
          - ☑ armeabi-v7a NEW
            - ☑ libagora-rtc-sdk-jni.so NEW
            - ☑ libagoraSdkCWrapper.so NEW

Note: Depending on your version of Unity, you may see and API Update Required alert window.

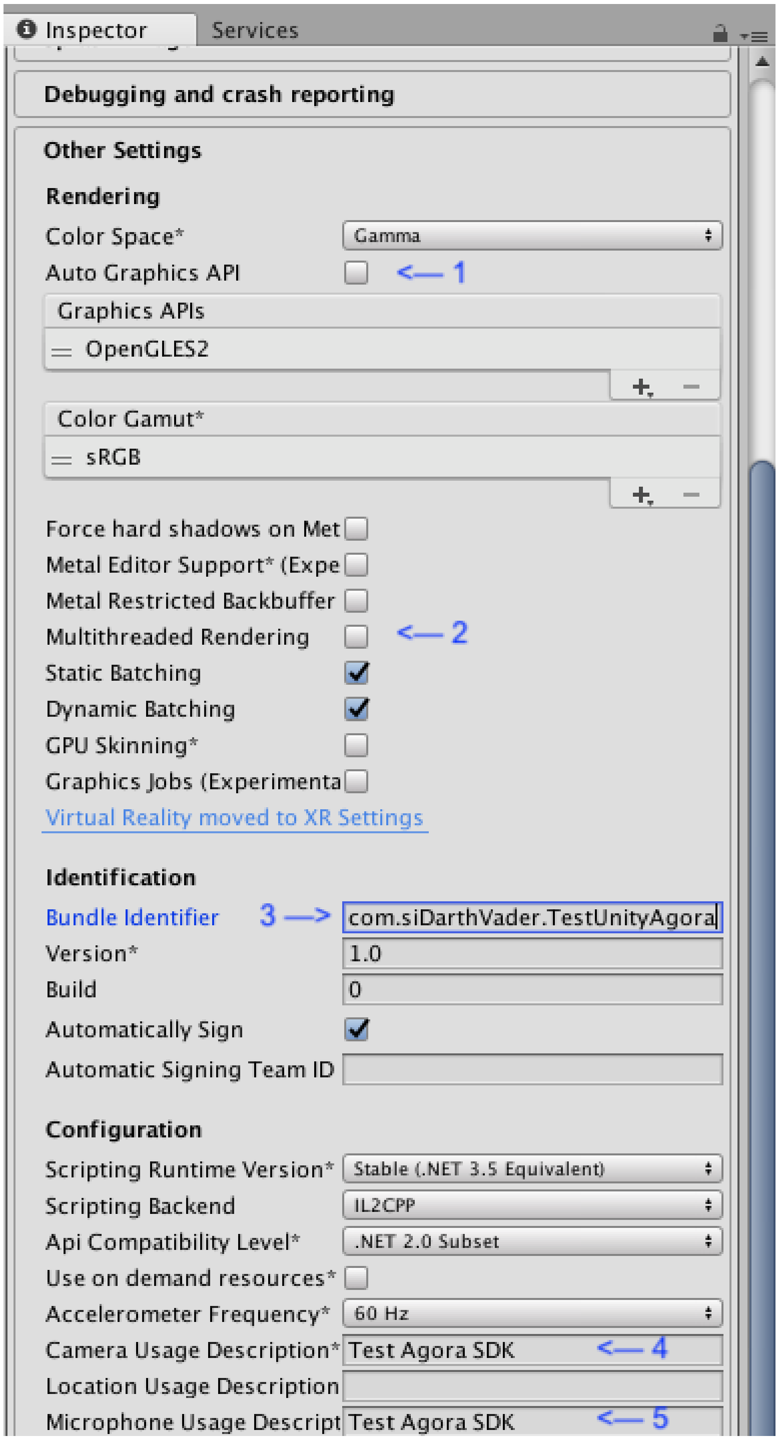
Once the assets are imported, double click the **TestSceneScript.cs** file. In the **TestSceneScript** class, update **YOUR APP ID** with your App ID.

```
private string appId = “YOUR APP ID”;
```

. . .

Running the Sample Application

Open the **Build Settings** and add the TestSceneHome.unity & TestSceneHomeWorld.unity scenes to the compilation and select the compilation platform (iOS)



Debugging and crash reporting

Other Settings

Rendering

Color Space* Gamma

Auto Graphics API □ <— 1

Graphics APIs

OpenGLES2

Color Gamut*

sRGB

Force hard shadows on Met □

Metal Editor Support* (Expe □

Metal Restricted Backbuffer □

Multithreaded Rendering □ <— 2

Static Batching ☑

Dynamic Batching ☑

GPU Skinning* □

Graphics Jobs (Experimenta □

Virtual Reality moved to XR Settings

Identification

Bundle Identifier 3 —> com.siDarthVader.TestUnityAgora

Version* 1.0

Build 0

Automatically Sign ☑

Automatic Signing Team ID

Configuration

Scripting Runtime Version* Stable (.NET 3.5 Equivalent)

Scripting Backend IL2CPP

Api Compatibility Level* .NET 2.0 Subset

Use on demand resources* □

Accelerometer Frequency* 60 Hz

Camera Usage Description* Test Agora SDK <— 4

Location Usage Description

Microphone Usage Descript Test Agora SDK <— 5

1. Open the **Player Settings** panel deselect the **Auto Graphics API** option**.** Within the **Graphics API** options that are visible, make sure that only **OpenGLES2** is on the list and remove any other APIs that appear on the list. **Note: You may need to add OpenGLES2 by using the + icon)**

2. Next, deselect the **Multithreaded Rendering** option**.**

3. Change the Bundle Identifier to your own Bundle identifier so XCode can properly codesign the application.

4. Ensure the microphone permission has a description to allow the user to know why the microphone is being accessed by the application

5. Ensure the camera permission has a description to allow the user to know why the camera is being accessed by the application

6. Once all the setup is complete click **Build and Run.** Once XCode opens up, enable auto-signing *(or sign the project)* and run the sample application on your iOS test device. **Note:** The sample app will not run in the simulator as there is no way to access the camera and microphone on a simulator.

. . .

Other Resources

- A detailed code walkthrough for this sample is available within the Unity Asset under the Guide file.

- The complete API documentation is available in the Document Center.

- For technical support, submit a ticket using the Agora Dashboard.